Graph Embeddings from Random Neural Features

Daniele Zambon[1] **Cesare Alippi**[1,2] **Lorenzo Livi**[3,4]
[1]Università della Svizzera italiana, Switzerland. [2]Politecnico di Milano, Italy.
[3]University of Manitoba, Canada. [4]University of Exeter, United Kingdom.
daniele.zambon@usi.ch, cesare.alippi@usi.ch, lorenzo.livi@umanitoba.ca

Abstract

We present Graph Random Neural Features (GRNF), a novel embedding method from graph-structured data to real vectors based on a family of graph neural networks. The embedding naturally deals with graph isomorphism and preserves, in probability, the metric structure of graph domain. In addition to being an explicit embedding method, it also allows to efficiently and effectively approximate graph metric distances (as well as complete kernel functions); a criterion to select the embedding dimension trading off the approximation accuracy with the computational cost is provided. Derived GRNF can be used within traditional processing methods or as first layer of a larger neural network for graph-structured data. The theoretical guarantees that accompany GRNF ensure that the considered graph distance is metric, hence allowing to distinguish any pair of non-isomorphic graphs.

1 Introduction

Inference on graph-structured data is one of the hottest topics in machine learning, thanks to successes achieved in several scientific fields, like neurosciences, chemistry, computational biology and social sciences [1, 2, 3]. One of the major research challenges there consists in building a practical solution able to process graphs, yet managing the graph isomorphism problem. A way to address this latter problem passes through metric distances and complete kernels, however, it has been shown to be at least as hard as deciding whether two graphs are isomorphic [4].

When data come as real vectors, the seminal paper by Rahimi and Recht [5] provides an efficient method to approximate radial basis kernels, exposing a parameter —the embedding dimension— trading off computational complexity with approximation error. The Random Kitchen Sinks [6] technique builds on [5] by adding a linear trainable layer and achieves, via convex optimization, an optimal estimation error on regression tasks [7]; see also [8, 9] for discussions. More recently, significant efforts aim at moving the research to the graph domain [10], with most of the contributions focusing on graph neural network properties, especially on their ability to discriminate between non-isomorphic graphs [11, 12, 13]. Recent research has also provided neural architectures granting the universal approximation property for functions on the graph domain [14, 15]; this holds asymptotically with the number of neurons.

Here we propose Graph Random Neural Features (GRNF), an embedding method that provides a map $\mathbf{z} : \mathcal{G} \to \mathbb{R}^M$ from attributed graphs to numeric vectors, and manages the graph isomorphism problem. We prove that GRNF is able to discriminate between any pair of non-isomorphic graphs in probability and preserves, approximately, the metric structure of the graph domain in the embedding space. Furthermore, GRNF can also be employed as first layer of a larger graph neural network architecture. What proposed applies to graphs with non-identified nodes, meaning that there is no one-to-one correspondence between nodes of different graphs, and no information is brought by the order of the nodes. The graphs can be directed or undirected, with variable (yet, finite) order and topology, and scalar attributes associated with nodes and edges. We refer the reader to [16] for further details, proofs and extended experimental evaluation.

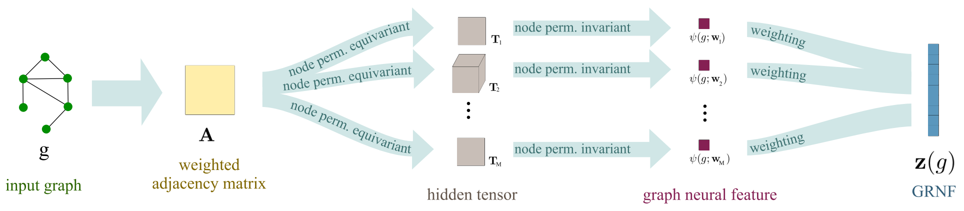

Figure 1: Scheme of a GRNF map $\mathbf{z} : \mathcal{G} \to \mathbb{R}^M$. A n-node graph $g \in \mathcal{G}$ is firstly represented as a weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{n^2}$. A collection of M graph neural features $\{\psi(g; \mathbf{w}_m)\}_{m=1}^M$ is then computed, weighted and, finally, concatenated in vector $\mathbf{z}(g)$. As described in Section 2.1, each graph neural feature map $\psi(\,\cdot\,; \mathbf{w}_m)$ is the composition of a node-permutation equivariant function, that maps matrix $\mathbf{A}$ to a tensor $\mathbf{T}_m \in \mathbb{R}^{n^k}$ of order k, and a node-permutation invariant one, that maps $\mathbf{T}_m$ to a scalar value $\psi(g; \mathbf{w}_m) \in \mathbb{R}$. Notice that order of tensors $\{\mathbf{T}_m\}_{m=1}^M$ needs not to be the same; in this visual example $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_M$ have orders $2, 3$ and 2, respectively.

2 Proposed method

As shown in Figure 1, the GRNF architecture is built upon graph features $\psi(\,\cdot\,; \mathbf{w}) : \mathcal{G} \to \mathbb{R}$ that we called "graph neural features". Suitably, sampling and weighting these features, results in the proposed embedding map $\mathbf{z}(\cdot)$.

2.1 Graph Neural Features

A *graph neural feature* is a parametric scalar function $\psi(\,\cdot\,; \mathbf{w})$ resulting from the composition of:

- An affine layer $F_k(\,\cdot\,; \boldsymbol{\theta}_F) : \mathcal{G} \to \mathbb{R}^{n^k}$ that maps graph g to a tensor of order k; $F_k(\,\cdot\,; \boldsymbol{\theta}_F)$ is equivariant, meaning that permuting the node ordering in g results in the same tensor with the permutation applied to each of its dimensions.
- A continuous squashing function $\rho_e : \mathbb{R} \to \mathbb{R}$, applied component-wise.
- An affine layer $H_k(\,\cdot\,; \boldsymbol{\theta}_H) : \mathbb{R}^{n^k} \to \mathbb{R}$ that maps a tensor of order k to a scalar value; $H_k(\,\cdot\,; \boldsymbol{\theta}_H)$ needs to be invariant under node permutations, meaning that independently on the node permutation of input tensor, the output will always be the same.
- A non-constant continuous function $\rho_i : \mathbb{R} \to \mathbb{R}$.

It has been proven by Maron et al. [17] that affine invariant and equivariant functions for graphs form a vector space of dimensions $\mathrm{b}(k) + 1$ and $\mathrm{b}(k + l) + \mathrm{b}(k)$, respectively; $\mathrm{b}(k)$ here denotes the number of partitions of set $\{1, \dots, k\}$. For this reason, $H_k(\,\cdot\,; \boldsymbol{\theta})$ and $F_k(\,\cdot\,; \boldsymbol{\theta})$ can be written in terms of tensor products and sums of the form

$$H_k(\mathbf{T}; \boldsymbol{\theta}) = \sum_{\gamma=1}^{\mathrm{b}(k)} \theta_\gamma \mathbf{I}_\gamma \cdot \mathbf{T} + \theta'_0, \qquad F_k(g; \boldsymbol{\theta}) = \sum_{\gamma=1}^{\mathrm{b}(k+2)} \theta_\gamma \mathbf{E}_\gamma \cdot \mathbf{A}_g + \sum_{\gamma'=1}^{\mathrm{b}(k)} \theta'_{\gamma'} \mathbf{I}_{\gamma'}$$

where $\mathbf{A}_g \in \mathbb{R}^{n^2}$ is a matrix representation of n-node graph g, and $\{\mathbf{E}_\gamma\}_{\gamma=1}^{\mathrm{b}(k+2)} \subseteq \mathbb{R}^{n^k+n^2}$ and $\{\mathbf{I}_\gamma\}_{\gamma=1}^{\mathrm{b}(k)} \subseteq \mathbb{R}^{n^k}$ are the basis tensors of the equivariant and invariant linear spaces, respectively. Finally, $\boldsymbol{\theta} = \{\theta_\gamma\} \cup \{\theta'_{\gamma'}\}$ stores function parameters. Please, refer to the paper by Maron et al. [17] for a detailed description. The parameter vector of the resulting graph neural feature $\psi(\,\cdot\,; \mathbf{w})$ is $\mathbf{w} = (k, \boldsymbol{\theta}_F, \boldsymbol{\theta}_H)$, which also encodes the order k of hidden tensor $\mathbf{T}$.

We denote with $\mathcal{F} = \{\psi(\,\cdot\,; \mathbf{w}) \mid \mathbf{w} \in \mathcal{W}\}$ the family of graph neural features on parameter space

$$\mathcal{W} = \left\{ \mathbf{w} = (k, \boldsymbol{\theta}_F, \boldsymbol{\theta}_H) \quad \text{s.t.} \quad k \in \mathbb{N},\ \boldsymbol{\theta}_F \in \mathbb{R}^{\mathrm{b}(k+2)+\mathrm{b}(k)},\ \boldsymbol{\theta}_H \in \mathbb{R}^{\mathrm{b}(k)+1} \right\}. \tag{1}$$

Notice that despite we defined the features for a fixed number n of nodes, parameter $\mathbf{w} \in \mathcal{W}$ are independent from n and the same definition can be readily extended to graph of arbitrary size.

Given a probability function p on set $\mathcal{W}$, a distance between graphs can[1] be defined:

$$d_p(g_1, g_2) = \left(\mathbb{E}_{\mathbf{w} \sim p} \left[(\psi(g_1; \mathbf{w}) - \psi(g_2; \mathbf{w}))^2 \right] \right)^{\frac{1}{2}}, \qquad \forall g_1, g_2 \in \mathcal{G}, \tag{2}$$

[1]This is possible whenever distribution p is such that $\mathbb{E}_{\mathbf{w}}[\psi(g; \mathbf{w})^2] < \infty$ for every $g \in \mathcal{G}$.

which assesses the expected discrepancy between graph neural features. In a supervised setting, we will see that distribution p is a "parameter" that can be tuned. In general, however, we prove that one good choice is to take p such that its support $\mathrm{supp}(p)$ is the entire set $\mathcal{W}$.

Theorem 1. *If the support* $\mathrm{supp}(p)$ *of* p *covers* $\mathcal{W}$*, then the space* $(\mathcal{G}, d_p)$ *is metric.*

Sketch of the proof. The core of the proof consists of showing that d_P enjoys the identifiability property:

$$\text{for any } g_1, g_2 \in \mathcal{G}, \qquad g_1 = g_2 \iff d_p(g_1, g_2) = 0. \tag{3}$$

To prove that it, we first need the following lemma.

Lemma 1. *Set* $\mathcal{F}$ *separates*[2] *graphs of* $\mathcal{G}$.

To prove Lemma 1 one can follow Hornik et al. [18] and Keriven and Peyré [15].

Getting back to Theorem 1, Lemma 1 ensures that there exists a particular $\tilde{\mathbf{w}} \in \mathcal{W}$ for which $\psi(g_1; \tilde{\mathbf{w}}) \neq \psi(g_2; \tilde{\mathbf{w}})$. Because $\mathrm{supp}(p) = \mathcal{W}$, and together with the continuity of ρ_i, ρ_e, we can find a neighbourhood $U(\tilde{\mathbf{w}})$ of non-null probability where $\psi(g_1; \mathbf{w}) \neq \psi(g_2; \mathbf{w}), \forall \mathbf{w} \in U(\tilde{\mathbf{w}})$; this ensures that the expectation in (2) is strictly positive, and hence property (3) holds. □

2.2 Graph Random Neural Features

The definition of graph distance d_p in (2) as an expectation over the feature maps in $\mathcal{F}$ allows us to create a random mapping that approximately preserves the metric structure of graph space $(\mathcal{G}, d_p)$.

Definition 1 (Graph Random Neural Features (GRNF)). *Given probability distribution* p *defined over* $\mathcal{W}$ *and an embedding dimension* $M \in \mathbb{N}$*, we define* Graph Random Neural Features map *a function* $\mathbf{z} : \mathcal{G} \to \mathbb{R}^M$ *that associates a graph* $g \in \mathcal{G}$ *with vector*

$$\mathbf{z}(g) := \mathbf{z}(g; \mathbf{W}) := \frac{1}{\sqrt{M}} \left[\psi(g; \mathbf{w}_1), \dots, \psi(g; \mathbf{w}_M)\right]^\top \tag{4}$$

where $\mathbf{W} = \{\mathbf{w}_m\}_{m=1}^M$ *are drawn independently from* p.

By sampling from a different distribution $\overline{p}$, we define also the *weighted-GRNF* map

$$\overline{\mathbf{z}}(g; \mathbf{W}) := \tfrac{1}{\sqrt{M}} \left[\dots, \sqrt{\tfrac{p(\mathbf{w}_m)}{\overline{p}(\mathbf{w}_k)}}\psi(g; \mathbf{w}_m), \dots\right]^\top. \tag{5}$$

Along the same lines of random Fourier features [5, 19] —but focusing on a distance rather than a kernel— we have that the squared norm $|\mathbf{z}(g_1) - \mathbf{z}(g_2)|_2^2$ reads as the sample mean $\frac{1}{M}\sum_m(\psi(g_1, \mathbf{w}_m) - \psi(g_2, \mathbf{w}_m))^2$, hence it results in an unbiased estimator of the squared distance $d_P(g_1, g_2)^2$. Moreover, by the law of large numbers, we have the following convergence

$$|\mathbf{z}(g_1) - \mathbf{z}(g_2)|_2^2 \to d_P(g_1, g_2)^2, \text{ in probability as } M \to \infty. \tag{6}$$

We prove the following theorem.

Theorem 2. *When there exists a constant* C *such that* $\mathbb{E}[\psi(g; \mathbf{w})^4] < C$ *uniformly in* $g \in \mathcal{G}$ *(w.l.o.g.* $C = 1$*), then for any value of* $\varepsilon > 0$ *and* $\delta \in (0, 1)$,

$$\mathbb{P}\left(\left||\mathbf{z}(g_1) - \mathbf{z}(g_2)|_2^2 - d_P(g_1, g_2)^2\right| \geq \varepsilon\right) \leq \delta,$$

with a choice of $M \geq \frac{16}{\delta\,\varepsilon^2}$*. The same holds also for the weighted-GRNF map* $\overline{\mathbf{z}}$.

Sketch of the proof. Being x^p convex, by the Jensen's inequality we have $\frac{1}{2^p}\mathbb{E}\left[(X+Y)^p\right] = \mathbb{E}\left[\left(\frac{X}{2} + \frac{Y}{2}\right)^p\right] \leq \frac{1}{2}(\mathbb{E}[X^p] + \mathbb{E}[Y^p])$. Specializing for $p = 4$ we get,

$$\begin{aligned}\mathrm{Var}\left[|\mathbf{z}(g_1) - \mathbf{z}(g_2)|_2^2\right] &= \frac{1}{M^2}\sum_{m=1}^M \mathrm{Var}\left[\left(\psi(g_1; \mathbf{w}) - \psi(g_2; \mathbf{w})\right)^2\right] \\ &\leq \frac{8}{M}\left(\mathbb{E}\left[\psi(g_1; \mathbf{w})^4\right] + \mathbb{E}\left[\psi(g_2; \mathbf{w}^4\right]\right) \leq \frac{16}{M}.\end{aligned}$$

The thesis follows by exploiting Chebyshev's inequality. □

[2]This means that for any pair of distinct (non-isomorphic) graphs $g_1, g_2 \in \mathcal{G}$ there exists a parameter vector $\mathbf{w} \in \mathcal{W}$ so that $\psi(g_1, \mathbf{w}) \neq \psi(g_2, \mathbf{w})$.

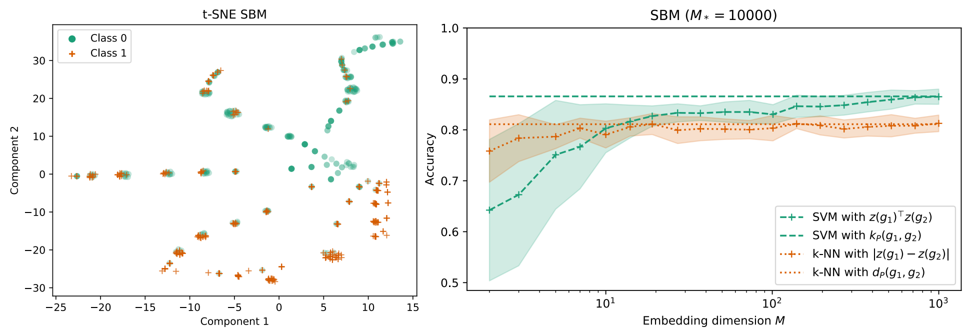

Figure 2: Left) 2-dimensional visualization of the embedding vectors $\mathbf{z}_*(g_i)$ for $i = 1, \dots, 600$ drawn with t-SNE. Right) Classification performance in terms of accuracy using different embedding dimensions M. Each reported accuracy value is an average across 10 repetitions and the shaded region represents one standard deviation around the average.

In practice, the specific choice of distribution p can make the difference. When working in a supervised setting, we can exploit the weighted-GRNF (5) to sample from a predefined distribution $\overline{p}$, and, a posteriori, identify a suitable distribution p that best serves the task to be solved [20]. Specifically, once M parameter vectors $\{\mathbf{w}_m\}_{m=1}^M$ are sampled from $\overline{p}$, we have that the scalars $\{p_m = p(\mathbf{w}_m)\}_{m=1}^M$ in (5) become tunable parameters.

A similar idea could also be pursued for a graph kernel of the form

$$\kappa_p(g_1, g_2) = \mathbb{E}_{\mathbf{w}}\left[\psi(g_1; \mathbf{w})\psi(g_2; \mathbf{w})\right],$$

which is intimately related with distance d_p. We can prove that $\kappa_p(g_1, g_2)$ is complete, meaning that the canonical embedding ϕ associated reproducing kernel Hilbert space is injective, and the same convergence of $\mathbf{z}(g_1)^\top\mathbf{z}(g_2)$ to $\kappa_p(g_1, g_2)$ holds, as $M \to \infty$.

3 Experimental validation

We compared the performance drop in adopting the approximation (6) with varying embedding dimension M. The considered task is binary classification, and it is solved using k-nearest neighbour classifier, as a standard distance-based methods. We generated two classes of 300 8-node graphs from the stochastic block model [21]. Class 0 has a single community with edge probability 0.4, while class 1 has two communities of 4 and 4 nodes with intra-community edge-probabilities of 0.8 and 0.6, respectively, and inter-community probability equal to 0.2.

As the exact computation of graph distance is unfeasible, we considered a M_*-dimensional map $\mathbf{z}_*$, with very large M_*, and considered $|\mathbf{z}_*(g_1) - \mathbf{z}_*(g_2)|_2$ to be equal to $d_p(g_1, g_2)$; similarly, for kernel k_p. Figure 2 shows a t-SNE [22] visualization of the data set and the classification performance achieved. We see that the classification accuracy obtained with GRNF empirical converges to the accuracy obtained with $M_* = 10^4$ features.

4 Related work and concluding remarks

The process of random sampling features takes inspiration from the work on random Fourier features [5], for which extensions have been proposed [23, 24], with some of them considering also graph data [25].

Our work operates somehow in the opposite direction. Firstly, it defines a parametric family of features $\mathcal{F}$ that separates graphs of $\mathcal{G}$ and, as a byproduct, it provides a distance function by selecting a convenient probability distribution p on parameter space $\mathcal{W}$. This choice has two major advantages: it does not require to compute distribution p as the Fourier transform (of the signature) of kernel κ, and it allows to search for the best distribution for the task at hand.

References

[1] Daniel C Elton, Zois Boukouvalas, Mark D Fuge, and Peter W Chung. Deep learning for molecular design–a review of the state of the art. *Molecular Systems Design & Engineering*, 2019. doi: 10.1039/C9ME00039A.

[2] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[3] Aming Li, Sean P Cornelius, Y-Y Liu, Long Wang, and A-L Barabási. The fundamental advantages of temporal networks. *Science*, 358(6366):1042–1046, 2017. doi: 10.1126/science.aai7488.

[4] Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, pages 129–143. Springer, 2003.

[5] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008.

[6] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*, pages 1313–1320, 2009.

[7] Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561. IEEE, 2008.

[8] Jose C Principe and Badong Chen. Universal approximation with convex optimization: Gimmick or reality?[discussion forum]. *IEEE Computational Intelligence Magazine*, 10(2):68–77, 2015.

[9] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

[10] Luca Oneto, Nicolò Navarin, Michele Donini, Alessandro Sperduti, Fabio Aiolli, and Davide Anguita. Measuring the expressivity of graph kernels through statistical learning theory. *Neurocomputing*, 268:4–16, 2017.

[11] Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. *arXiv preprint arXiv:1905.12560*, 2019.

[12] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. *arXiv preprint arXiv:1905.11136*, 2019.

[13] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

[14] Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. In *International Conference on Machine Learning*, pages 4363–4371, 2019.

[15] Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. *arXiv preprint arXiv:1905.04943*, 2019.

[16] Daniele Zambon, Cesare Alippi, and Lorenzo Livi. Distance-preserving graph embeddings from random neural features. *arXiv preprint arXiv:1909.03790*, 2019.

[17] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Syx72jC9tm`.

[18] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[19] Zhu Li, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards a unified analysis of random Fourier features. In *International Conference on Machine Learning*, pages 3905–3914, 2019.

[20] Aman Sinha and J C Duchi. Learning kernels with random features. In *Advances in Neural Information Processing Systems*, pages 1298–1306, 2016.

[21] P W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983. doi: 10.1016/0378-8733(83)90021-7.

[22] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov.):2579–2605, 2008.

[23] Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. In *Artificial Intelligence and Statistics*, pages 583–591, 2012.

[24] Felix Xinnan Yu, Ananda Theertha Suresh, Krzysztof Choromanski, Daniel Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pages 1975–1983, 2016.

[25] Lingfei Wu, I E-H Yen, Fangli Xu, Pradeep Ravikumar, and Michael Witbrock. D2ke: From distance to kernel and embedding. *arXiv preprint arXiv:1802.04956*, 2018.